

基本说明

通过utf-8编辑器实现功能, *notepad++* 可以支持列操作, 可用于定义同属性多个变量的时候 utf-8编辑器具体用于编辑操作界面的配置文件, 在其他循环调用下生成自定义的操作界面, 或直接在控制系统上创建对话框

现在支持两种语法结构 一种的传统句法--类文档路径 一种的扩展语法--类html,css

基本的功能

1. 对话框

1. 显示对话框: 提供软键, 变量, 文本以及帮助文本, 图形和帮助画面

2. 调用对话框: 调用的方法: 按下(登录)软键 / 选择PLC/NC

3. 动态重组对话框: 修改, 删除软键, 定义并设计变量栏, 显示, 更换, 删除文本以及图形

4. 对话框之间的数据交换
2. 功能

1. 变量操作: 读取, 写入, 变量的处理, 运算

2. 系统功能: 执行子程序, 文件功能, PI服务

3. 触发操作: 进行特定操作时自动触发的动作, 显示对话框, 输入值, 按下按键, 关闭对话框, 变量的值的变化

文件与编码

run myscreen 会运行在828或者840上面, 可能包含两种操作系统(linux,windows), 文件名, 后缀需要小写

文件

文件类型

1. com文件: 描述用户界面的所有内容以及实现的具体功能--程序文件
2. ini: 描述用户界面如何嵌入828/840系统中, **嵌入的方式以及相关功能的定义**--配置文件, 说明.com文件在哪里生效
3. 语言文件: 为txt, 在不同环境下显示不同文字字符串的文件(怀疑会放得到处都是, 没有集成的统一管理语言)
4. 图片文件: 为png,根据对应的屏幕分辨率放到不同的文件夹下
5. .html: 在线帮助源码, 按下按键之后弹出帮助文档

文件放置位置

一般的配置文件有4种文件夹

1. 系统西门子目录
2. 系统oem目录
3. 系统用户目录
4. 系统插件目录 优先级: 用户>oem(制造商)>插件>西门子 优先级含义, 如果多个文件夹中有同样的文件, 只会执行优先级高的目录里面的(会覆盖优先级低的) 放置的位置
5. run myscreen配置(.com):系统oem/proj
6. 配置文件: 系统/用户/oem /sinumerik /hmi / cfg
7. 语言文件: 系统/用户/oem /sinumerik /hmi / lng
8. 图片文件: 系统/用户/oem /sinumerik /hmi / ico[分辨率]
9. 在线帮助: 系统/用户/oem /sinumerik /hmi / help[语言]

日志

当编译文件出错时, 会将错误保存在用户目录/log下的\_log.txt中, 系统只会记录当前选中的对话框的错误, 之前选中的会被删除 包含的信息

1. 执行到哪一行出错
2. 第一个错误字符的行与列编号
3. 配置文件中所有的出错行
4. DEBUG生成的记录

启动入口 / 操作区

有专门的run myscreen的入口/启动区 各个操作面板下都有特殊位置的按钮可以进入, 还可以通过custom(定制)进入

配置软键的操作区/入口

通过easyscreen.ini文件来实现 针对从不同的入口有不同的计入的入口配置语句, 但大体上语法是一样的(看一看手册吧)

界面的创建及实现

配置文件之后, 就是实现源文件的代码(.com) 通常一个项目的基本框架是由**一组描述登入软键栏的代码及多组描述各幅页面的代码**组成 定义的语法查看手册

1. 描述登录软键栏 启动软键的定义等 使用软键定义的扩展语法比较好-类似css的语法 系统也提供了一些预定义的软键列表(类似宏定义), 查看手册即可
2. 各个页面的描述 页面的组成部分/框架-即需要考虑, 或者需要重新定义的部分

1. 状态显示栏--目前的操作状态

2. 页面的标题

3. 右侧8个垂直软键

4. 下面8个水平软键

5. 显示注释-可以指定这里的文本

6. 中间的页面区-可以包含图片显示单元, 文本显示单元(均是通过变量的形式实现) 关于页面的大概语法: 可以分成3个区域, 都包含在页面定义区里面

1. 页面元素定义

2. 页面软键定义

3. 页面动作响应 依然推荐使用扩展写法 装载屏幕窗口, 切换对话框

其他功能 从对话框1(主对话框)跳转至对话框2, 且带变量传输 切换对话框有两种情况, 一是会关闭主对话框, 二是不会关闭, 区别在于关闭对话框2时会返回到哪个对话框, 在装载对话框的时候决定-LM(.0)--已经验证了语法的正确性 默认是0, 关闭当前页面再装载新页面--运行exit时会退出到上一个页面, 而不是进入新页面的这个页面, 1是中断当前页面, 直接装载新页面--运行exit时会回到到新页面的这个页面

当在子页面定义背景图片的时候, 需要定义位置, 否则在后面定义变量的时候会无法显示

调用界面的方式有两种

1. 平时的写法
2. 将定义的界面写在另外一个同文件夹下的.com文件中, 当调用屏幕的时候的语法更改为

```
LM("MASK1", "MASK1.com")
```

MASK1.com为定义该界面的文件名

## 软键的定义

具体看手册，但需要注意的是 设置的属性可以在运行程序的时候更改 注意带图标的软键的分辨率

## 变量

### 定义

注意，定义变量的时候看预设值是关联用户变量或者系统变量还是我们自己赋值，这个赋值的前缀语法不一样，一个是VAR=,一个是VAL= 系统变量看nx手册

变量有2种 一个是HMI变量 一个是系统变量

变量的值也可以来自输入面板上的按键-各自有地址

界面中需要使用的文字，图片，输入框，局部变量等各种元素都需要定义 变量可以实现窗体元素的显示，值的运算

可以与系统变量相关联--如何关联？系统变量是什么？

具体的定义语法查看手册，与css差不多 推荐扩展语法

有一种特殊的变量，只对变量进行定义，不进行属性的指定，那么该变量不可见而且任意类型的变量都可以对其赋值

如果只想显示图片或者文字，可以不定义变量的属性，直接将短文本的属性填入即可

### 变量的显示方式

- 1. 文本+文本框
- 2. 下拉选择框
- 3. 列表
- 4. 静态图像显示-相当于将短文本赋值为图像---长/短文本是什么？-短文本是变量前面的解释，长文本是显示在状态栏里面的一个状态显示
- 5. 动态图像显示，可能是在输入框中进行赋值
- 6. 进度条
- 7. password
- 8. Qt Widget

### 变量的属性

常见的需要记住 长文本属性是显示在标题栏的右边的

### 文本，文本框的实现

#### 显示方式

确定类型 确定输入输出模式(可不可见)

### 下拉选择框的实现

转换值的概念

### 进度条的实现

需要将显示模式设置成1/2--可不可变色 设定前景色(进度条的颜色)与背景色(相当于进度条的框) 对于可变色的需要设定变色的阈值 对于颜色的限制，一个进度条最多显示4种颜色，包括背景色，需要显示更多的颜色需要多个进度条的拼接 设定进度条的显示选项，从哪个方向到哪个方向的进度

## 功能和方法

### 调试

DEBUG 可在屏幕设计阶段提供分析和帮助，只需在程序文件中输入特定语句即可 会显示时间 作用有点像printf("here")

DLGL 在对话框状态栏中根据给定的情况绘出短文本(信息或输入帮助)，有字符数的限制

### 子程序的定义及调用

类似于main与自己定义的函数 CALL 可以从方法的任何一个位置调用一个装载的子程序，允许嵌套 SUB 用于定义子程序

## 界面结构

### 主系统页面

即machine区 系统页面下面的7个按键叫做操作区 操作区1-6是系统的界面区 7是custom是用户独立的操作区--用户可以自行设计一整套自定义界面

### 操作树

是界面之间的跳转关系的说明 每一个操作界面都有固定的软键入口-跳转到子界面-子界面之间可以互相跳转

从上面以及手册给出的图示关系来看，‘子界面’是一个‘独立’的界面，相互之间没有级别之分

### 界面组成

即所有的子界面，系统界面的默认/可以支持的显示，操作元素

- 1. 标题栏(最上方)：机床状态显示
- 2. 界面变量：在操作界面中显示，是输入框，文本显示等‘元素’
- 3. 垂直，水平按键各8个
- 4. 对话框输出状态栏(对话框的下沿)
- 5. 界面变量：图形元素
- 6. 对话框-看到的操作界面的主体
- 7. 对话框标题栏(对话框的上沿)：显示标题和长文本

对话框

需要注意，对话框的x,y位置受到宽度的影响，貌似不设置宽度的大小，就不能控制到x,y位置

包含的元素/属性

- 1. 对话框名称--Mask1 用于系统识别
- 2. 标题--在进入到对话框界面之后显示在对话框上沿的标题-用于用户阅读
- 3. 尺寸
- 4. 变量
- 5. 图片
- 6. 属性
- 7. 系统或用户变量
- 8. 颜色

可以编辑的对话框属性(在方法动作中) 可读写的：标题，帮助画面，系统变量，背景色 只读的：界面的位置属性

语法结构

注意使用英文输入法 且开始定义的标记处是{}不是() 请按照下面的定义顺序书写，不然会报错 伪代码结构

```
界面开始标记
//M{此处设置界面的各种属性}
变量定义
DEF VAR1=
按键定义
H51=()
方法动作
LOAD
.....
END_LOAD

PRESS()
.....
END_PRESS
结束标记
//END
```

变量

对话框中显示的文本，列表框，输入框，图片等元素 通过 . 访问变量的各种属性--类似成员变量的访问

下拉选择框

注意属性的具体语法是 TGL="\*"X"";\*"Y"" ,表示下拉选择框的文本为X和Y，注意字符串需要两个双引号括起来 可以设定预设值，预设值就是下拉框中的某一个值

下拉选择框动态操作

可以查看功能，包括添加，删除元素

进度条

需要将DT属性设置为1(变色进度条) (其他的值对应其他不同的显示方式)-----注意与DT=2(双色进度条)区别，DT=2是方格形状的进度条 也需要注意确定该变量的最大值与最小值(用于系统确定进度条的范围) 前景色FC 背景色BC

双颜色的时候需要确定SC1,SC2属性的值---如果要显示多于2种颜色的话，需要将两个进度条拼接起来 确定变色范围为SVAL1,SVAL2

双颜色的时候其实进度条会有4种颜色，两个的变色，一个前景色，一个背景色 前景色是超过了SVAL2后显示的颜色

句法

类似css

可设置的变量参数(主要的)

- 1. 类型
- 2. 文本属性:
  - 1. 短文本：在对话框内以文本或者图形显示
  - 2. 长文本：在对话框标题栏显示的文本
  - 3. 提示文本
  - 4. 等等
- 3. 显示变量的位置，各种位置--用相应可使用的功能(短文本以及输入栏的位置--可以使用变量来改变位置--搭配检测变量的变化方法使用)

变量的赋值 (val)---(区别于var的用于关联系统/用户变量)

- 1. 预设值：定义时给出
- 2. 系统或者用户变量：进行关联
- 3. 直接赋值：界面方法中赋值

变量状态(vid)

在运行的时候可以查询该变量是否包含一个有效值 有的时候为true,反之为false

辅助变量

变量计算

在每次退出，输入/输出栏后计算变量--可以通过设置参数决定什么时候计算/更新变量的值

变量简介编译地址

就和c中用一个变量i访问数组的值一样 注意字符和变量之间用 << 连接

更改变量属性

通过 . 访问变量的各种属性直接等号赋值

按键

分为起始软键以及子界面的软件 两个的定义的位置有区别

调用软键栏的时候界面不会改变对话框

在定义软键之后才可以更改属性以及使用各种方法--加载子界面以及实现其他功能

定义软键栏

1. 定义起始软键栏
- //S (START)  
....  
//END
2. 定义子界面 在子界面的定义区域里面定义
- //M(mask1,HD="机器人管理")  
...  
//END

软键句法-定义属性

设置可见性等 注意适用扩展语法时适用3双引号

更改软件属性

可以在方法中更改的软键属性 通过 : 访问属性并且等号赋值

1. 文本
2. 访问等级--值=ac{n} n为等级
3. 状态
4. 等等

系统预定义的软键

直接使用类似宏定义的语句

触摸屏按钮

与按键不同，触摸屏按钮是用于触屏的按钮，可以放置在任何位置 有硬件要求 算是一种变量，可以实现函数操作 --之后再

文件说明

初始化文件 easyscreen.ini

用于声明界面入口文件，语言文件等

[STARTFILES]下定义界面(不同界面会实现不同的功能)所属位置及(实现程序的)文件名----- 这里的界面的意思是'操作区'吧？ 注意调试区域的叫做startup，参数区域叫做offset [LANGUAGEFILES]下定义语言文件的文件名 --可以参考模板文件书写

入口关联的.com文件

\*\*入口关联的.com文件定义的start界面不是初始的系统界面(AreaMachine)，而是点击进入对应操作区时的开始界面(custom或者jog等)\*\*也就是说，初始界面是machine区

要求:

1. 只能有一个起始软键栏
2. 仅在 //S 下定义入口按键位置(参考.ini中的空余软键位置--有要求)以及按下按键的方法
3. 对话框的定义可以在关联的.com中编写，也可以在另外一个.com中

slamgonfig.ini 文件

是一个配置文件 说明某个界面的默认配置。

1. 语言
2. 启动软键的图标
3. 启动软键的文字
4. 启动软键的位置
5. 需要的访问等级

语法

:或者 # 进行注释

方法

通过不同的事件触发某些特定的动作

基本编程方式

伪代码

- 方法的开始标记
- 功能
- 改变变量的属性
- 计算变量
- 方法的结束标记

LOAD

在加载该界面前就会运行的方法

具体的方法种类

1. 访问等级--在界面访问等级改变时，会自动运行 语法类似switch case 根据不同的访问等级(一个系统变量)执行不同操作
2. 变量值的改变 变量值改变的时候运行该方法 可以指定变量，或者界面中的任意一个变量改变时
3. 通道切换 界面通道切换的时候运行 由指定的系统变量记录目前的通道号。 根据切换到不同的通道号，可以执行不同的操作--需要if判断语句
4. 等等，自行查看手册

功能

类似系统已经定义的函数，使用‘函数名’可以直接调用，并且有相应的返回值 有以下功能可以实现

- 1. 读写驱动
- 2. 读写系统，用户变量-注意格式
- 3. 块，数组，子程序，表格定义
- 4. 程序文件--复制文件，写入内容.... 貌似无法在NC文件中进行操作
- 5. 文件读写等等

RNP 功能

读写NC,PLC变量 注意读取的输入数据的格式

DLGL功能

在对话框下方的显示文本

WNP功能

写入变量，注意变量的地址写法以及要重新启动才启用可以写入功能